Contents

Configure tracing using MeshConfig and Pod annotations

5 minute read page test

Users are encouraged to transition to the Telemetry API for tracing configuration.

Istio provides the ability to configure advanced tracing options, such as sampling rate and adding custom tags to reported spans. Sampling is a beta feature, but adding custom tags and tracing tag length are considered in-development for this release.

Before you begin

1. Ensure that your applications propagate tracing headers as described here.

2. Follow the tracing installation guide located under Integrations based on your preferred tracing backend to install the appropriate addon and configure your Istio proxies to send traces to the tracing deployment.

Available tracing configurations

You can configure the following tracing options in Istio:

1. Random sampling rate for percentage of requests that will be selected for trace generation.
2. Maximum length of the request path after which the path will be truncated for reporting. This can be useful in limiting trace data storage specially if you’re collecting traces at ingress gateways.
3. Adding custom tags in spans. These tags can be added based on static literal values, environment values or fields from request headers. This can be used to inject additional information in spans specific to your environment.

There are two ways you can configure tracing

options:

1. Globally via `MeshConfig` options.
2. Per-pod annotations for workload specific customization.

In order for the new tracing configuration to take effect for either of these options you need to restart pods injected with Istio proxies.

Any pod annotations added for tracing configuration override global settings. In order to preserve any global settings you should copy them from global mesh config to pod annotations along with workload specific customization. In particular, make sure that the tracing backend address is always provided in the annotations to ensure that the traces are reported correctly for the workload.

Installation

Using these features opens new possibilities for managing traces in your environment.

In this example, we will sample all traces and add a tag named `clusterID` using the `ISTIO_META_CLUSTER_ID` environment variable injected into your pod. Only the first 256 characters of the value will be used.

```
$ cat <<EOF > ./tracing.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        sampling: 100.0
        max_path_tag_length: 256
        custom_tags:
          clusterID:
          environment:
            name: ISTIO_META_CLUSTER_ID
EOF
$ istioctl install -f ./tracing.yaml
```

Using `MeshConfig` for trace settings

All tracing options can be configured globally via `MeshConfig`. To simplify configuration, it is recommended to create a single YAML file which you can pass to the `istioctl install -f` command.

```
cat <<'EOF' > tracing.yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        sampling: 10
        custom_tags:
          my_tag_header:
            header:
              name: host
EOF
```

Using `proxy.istio.io/config`

annotation for trace settings

You can add the `proxy.istio.io/config` annotation to your Pod metadata specification to override any mesh-wide tracing settings. For instance, to modify the `sleep` deployment shipped with Istio you would add the following to `samples/sleep/sleep.yaml`:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sleep
spec:
  ...
  template:
    metadata:
      ...
      proxy.istio.io/config: |
        tracing:
          sampling: 10
          custom_tags:
            my_tag_header:
              header:
                name: host
    spec:
      ...
```

Customizing Trace sampling

The sampling rate option can be used to control what percentage of requests get reported to your tracing system. This should be configured depending upon your traffic in the mesh and the amount of tracing data you want to collect. The default rate is 1%.

> Previously, the recommended method was to change the `values.pilot.traceSampling` setting

during the mesh setup or to change the `PILOT_TRACE_SAMPLE` environment variable in the pilot or istiod deployment. While this method to alter sampling continues to work, the following method is strongly recommended instead.

In the event that both are specified, the value specified in the `MeshConfig` will override any other setting.

To modify the default random sampling to 50, add the following option to your `tracing.yaml` file.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        sampling: 50
```

The sampling rate should be in the range of 0.0 to 100.0 with a precision of 0.01. For example, to trace 5 requests out of every 10000, use 0.05 as the value here.

Customizing tracing tags

Custom tags can be added to spans based on literals, environmental variables and client request headers in order to provide additional information in spans specific to your environment.

There is no limit on the number of custom tags that you can add, but tag names must be unique.

You can customize the tags using any of the three supported options below.

1. Literal represents a static value that gets added to each span.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        custom_tags:
          my_tag_literal:
            literal:
              value: <VALUE>
```

2. Environmental variables can be used where the value of the custom tag is populated from a workload proxy environment variable.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        custom_tags:
          my_tag_env:
            environment:
              name: <ENV_VARIABLE_NAME>
              defaultValue: <VALUE>      # optional
```

In order to add custom tags based on environmental variables, you must modify the `istio-sidecar-injector` ConfigMap in your root Istio system namespace.

3. Client request header option can be used to populate tag value from an incoming client request header.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        custom_tags:
          my_tag_header:
            header:
              name: <CLIENT-HEADER>
              defaultValue: <VALUE>      # optional
```

Customizing tracing tag length

By default, the maximum length for the request path included as part of the `HttpUrl` span tag is 256. To modify this maximum length, add the following to your `tracing.yaml` file.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    enableTracing: true
    defaultConfig:
      tracing:
        max_path_tag_length: <VALUE>
```